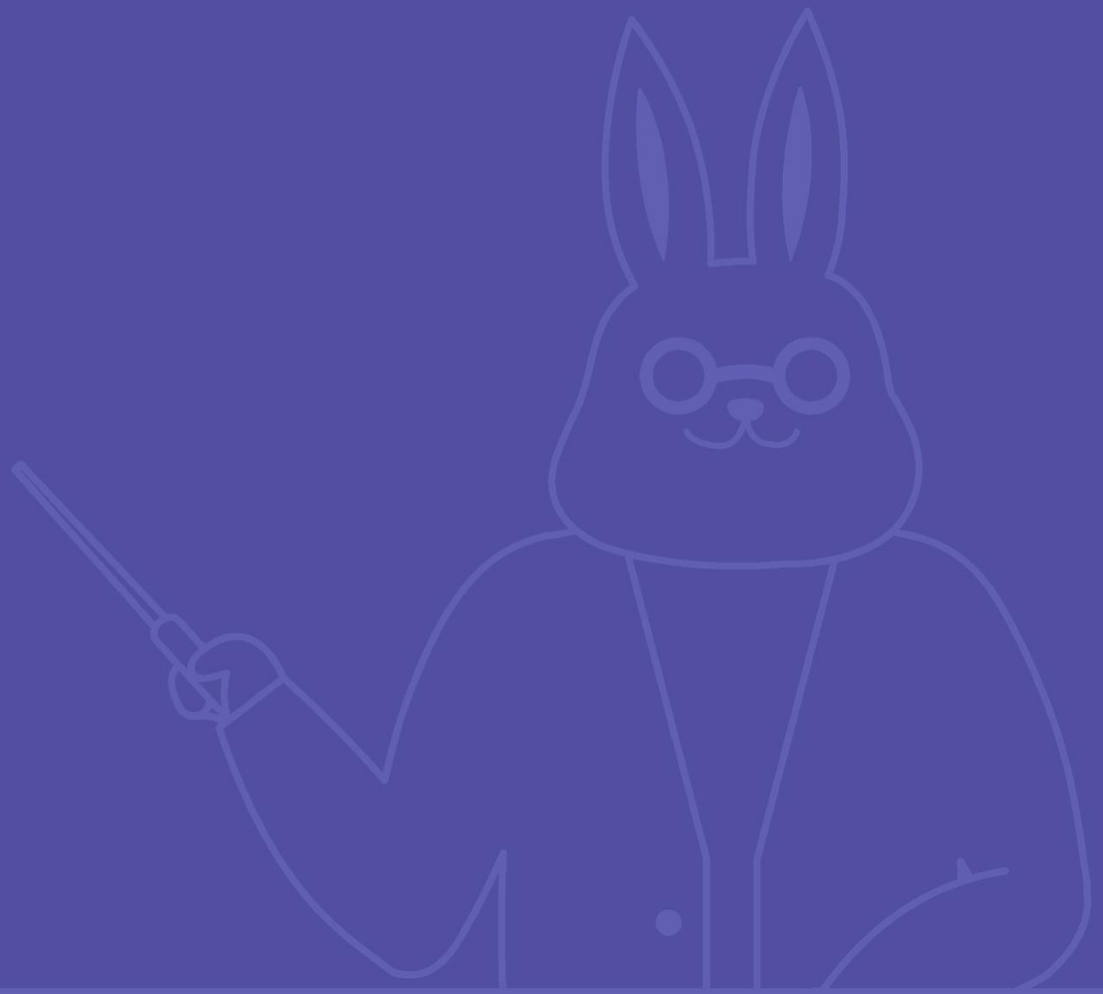
Flask 기초

03 RDB로 리소스 관리 및 저장하기



목차

🎯 수강목표

데이터베이스의 종류를 안다.

데이터베이스의 종류를 알고, 각각의 특징을 설명할 수 있습니다.

Flask에서 DB가 하는 역할을 이해한다.

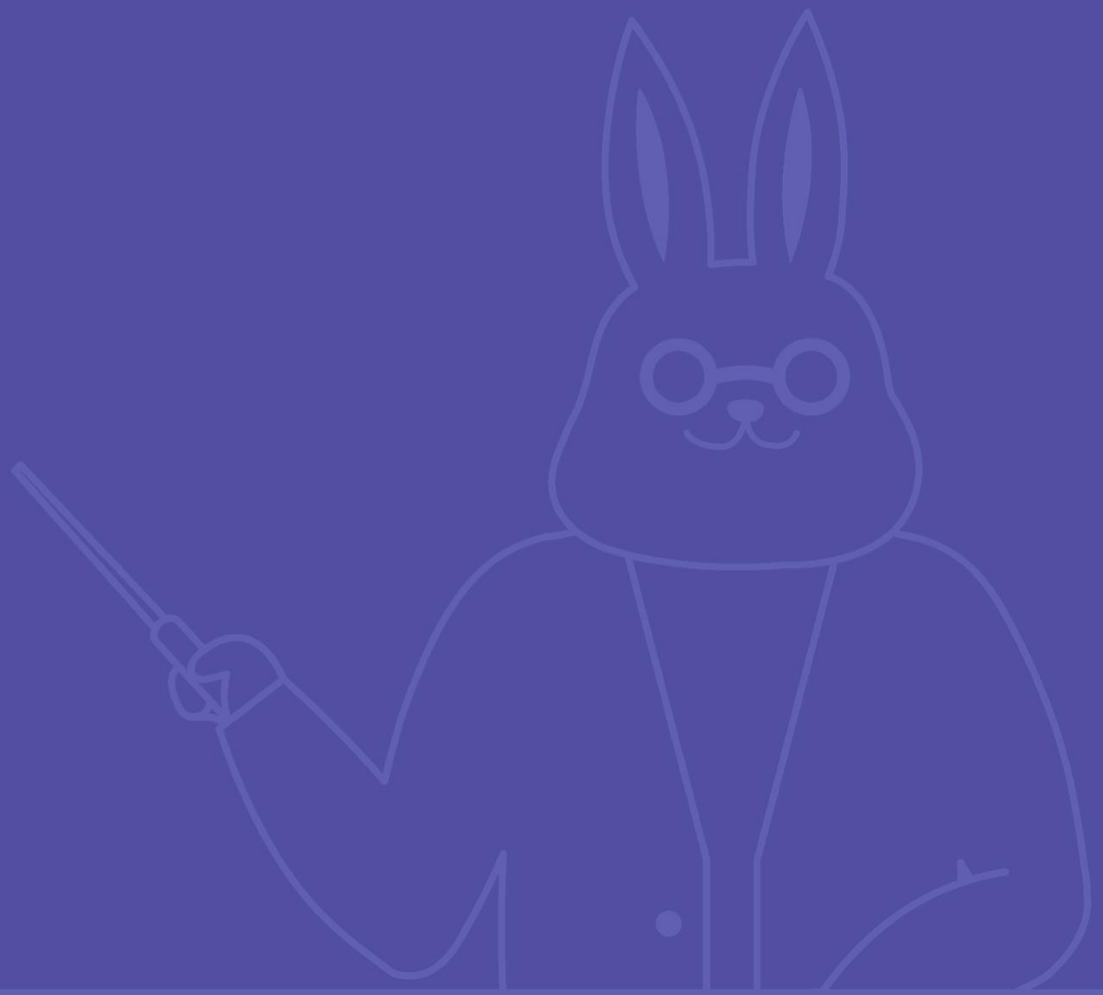
Flask에서 DB가 하는 역할을 이해하고, Database를 사용해서
서비스를 작성할 수 있습니다.

JWT를 이해 할 수 있다.

JWT의 개념을 이해하고, 사용하는 방법과 이유를 설명할 수 있습니다.

01

관계형 데이터베이스



데이터베이스

데이터베이스(Database)는 데이터를 저장하는 공간으로서, **서비스를 개발**하는 곳에서는 **빠질 수 없는 중요한 요소**입니다.

데이터베이스의 종류는 크게 관계형 데이터베이스(RDB)와 NoSQL(Not Only SQL)로 나누어져 있습니다.

관계형 데이터베이스(RDB)

키(key)와 **값(value)**들의 간단한 관계를 테이블 화 시킨 데이터베이스

RDB의 특징

DML을 사용해서 데이터 간 결합, 제약조건 등의 설정을 통해 데이터를 추출할 수 있습니다.

테이블 간의 데이터 관계를 설정할 수 있습니다.

RDB의 종류

Oracle
MySQL
MS-SQL
DB2
Maria DB
Derby
SQLite
...

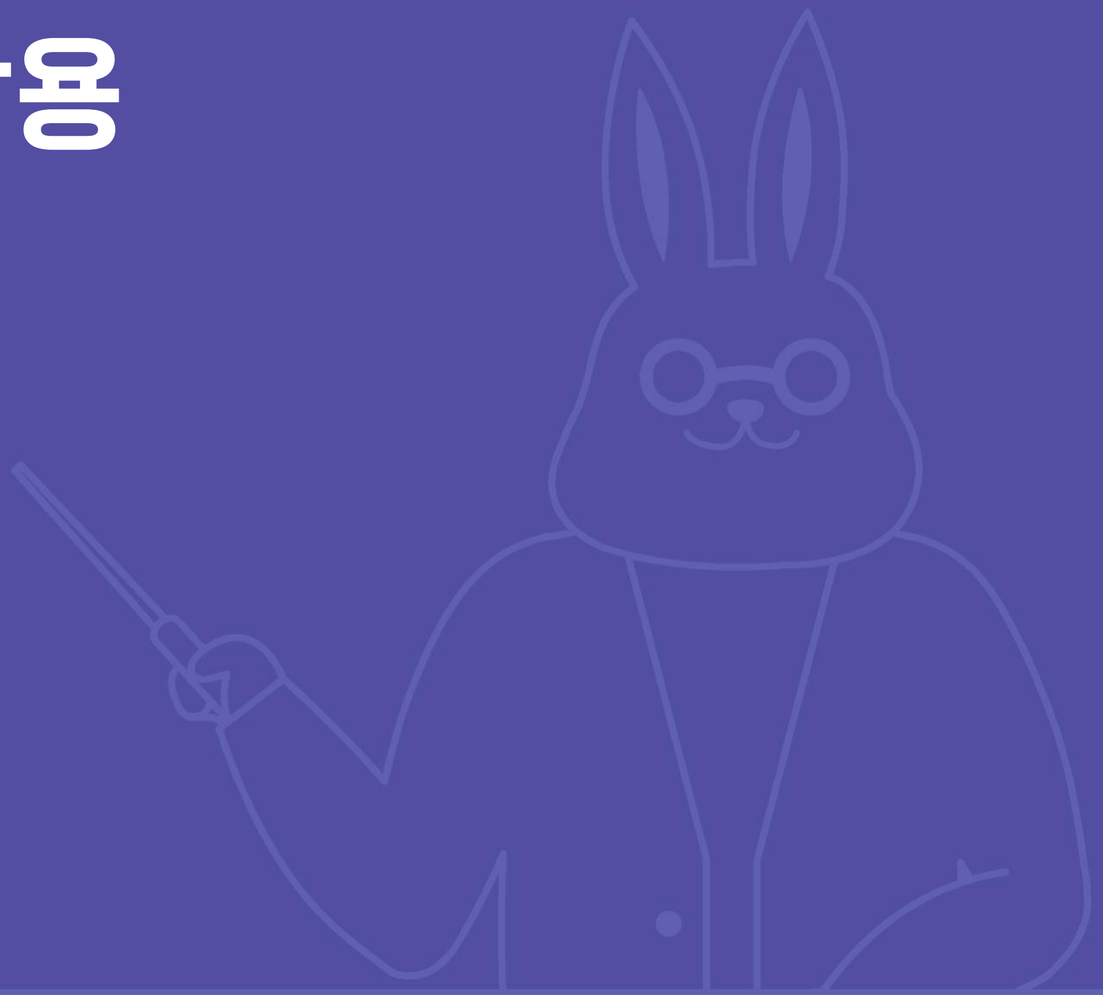
RDB의 형태

id	name	phone	address	birth	age
1	elice	0101111****	gangnam	9/23	27
2	oliver	0101234****	bundang	7/14	20
3	tom	0109876****	sokcho	6/18	22
4	tony	0109476****	tokyo	7/25	31

- RDB는 정형화된 데이터를 저장하고 있습니다. (다른 형태의 데이터가 들어올 수 없습니다)
- 각 컬럼(세로줄) 마다 데이터의 형태가 동일합니다.
- SQL 질의어 (SQL 쿼리)를 사용합니다.

02

RDB와 Flask의 상호작용

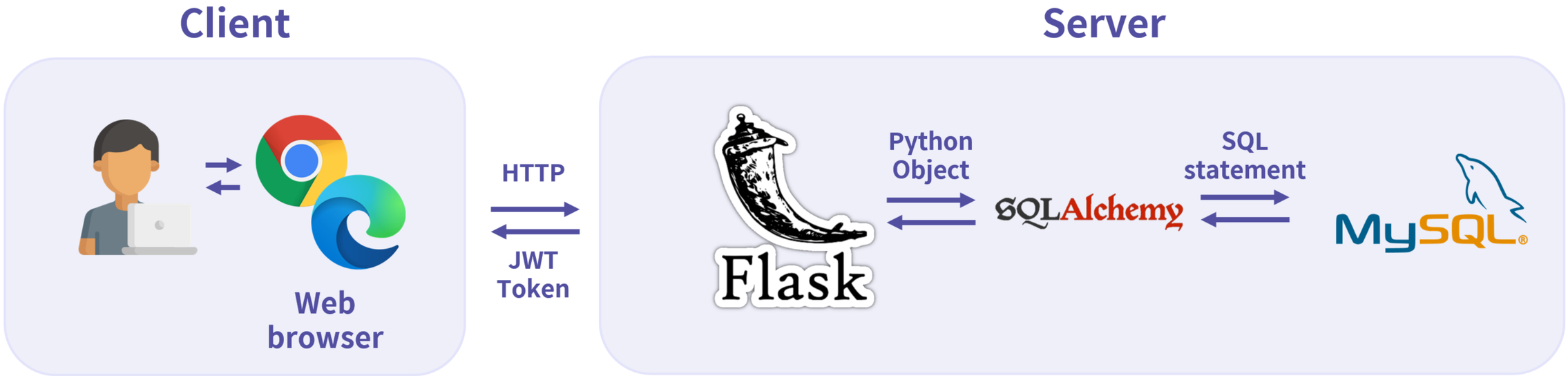


RDB와 Flask

파이썬은 오픈 소스와 상용 데이터베이스에 대한 대부분의 데이터베이스 엔진을 위한 패키지를 가지고 있습니다.

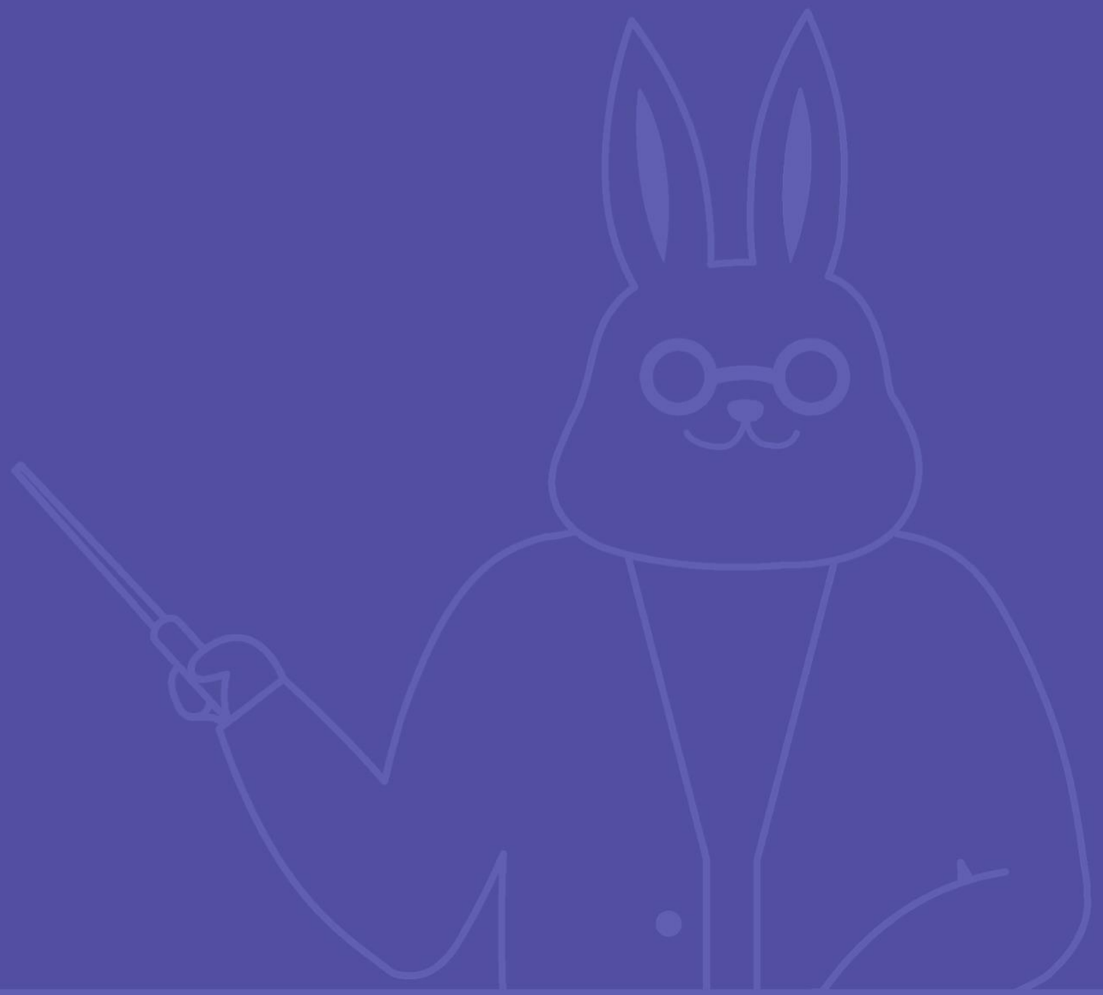
Flask에서 입력받은 내용을 DB에 저장할 수 있습니다.
→ 효율적인 **데이터 관리 가능**

RDB와 Flask


Client
Server
Web
browser
HTTP
JWT
Token
Flask
Python
Object
SQLAlchemy
SQL
statement
MySQL

03

간단한 게시판 만들기



구현 사항

데이터베이스에 저장되는 데이터를 활용해서 사용자를 검색해봅니다.

데이터베이스에 저장되는 데이터를 활용해서 사용자를 추가해봅니다. 단, 중복 사용자를 방지합니다.

게시판 내용을 생성, 조회, 수정, 삭제해봅니다.

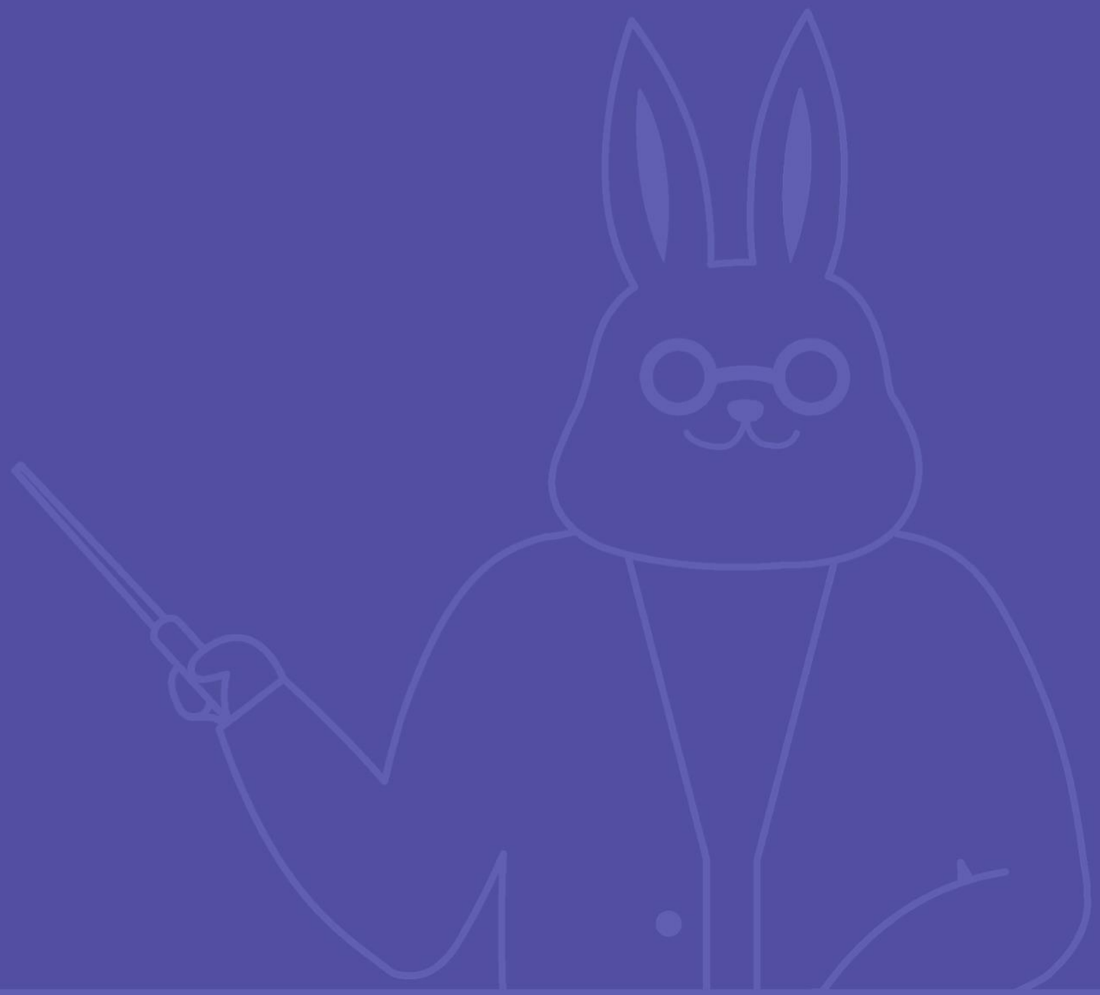
게시판 등록

게시판의 기능을 DB에 적용하기 위해서는 리소스를 다뤄야 합니다.

- 아이템 리소스를 데이터베이스에 **쓰기**(Create)
- 데이터베이스에서 아이템 리소스 **검색**(Read)
- 데이터베이스에서 아이템 리소스 **수정**(Update)
- 데이터베이스에서 아이템 리소스 **삭제**(Delete)

04

Flask JWT



JWT(JSON Web Token)란?

웹 표준(RFC 7519)으로서 두 개체에서 **JSON 객체**를 사용하여 통신합니다.

- JSON 포맷을 이용하여 사용자에 대한 속성을 저장하는 Web Token
- 토큰 자체를 정보로 사용하는 Self-Contained 방식으로 정보를 안정하게 전달

JWT의 생김새

header . payload . signature

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6ImVsaWNlIiwiaWF0IjoxNTE2MjM5MDIyfQ.KeQ_ZRON0VcycCQ6YtOIoNfKPvUjn8gNlR6HdE_xsis

Header에는 **토큰의 타입과 알고리즘**을 저장합니다.
Payload에는 토큰에 **담을 정보**를 넣습니다.
Signature에는 헤더와 정보의 **인코딩 값들과 관련된 비밀키**가 들어있습니다.

JWT 구조

JWT는 **Header, Payload, Signature** 3가지로 이루어지며, JSON 형태인 각 부분은 **Base64로 인코딩**되어 표현됩니다.

각각의 부분을 이어주기 위해 **.구분자를 사용하여 구분**합니다.

Base64는 암호화된 문자열이 아니고, 동일한 문자열에 대해 항상 같은 인코딩 문자열을 반환합니다.

Header

토큰의 헤더는 typ와 alg 총 두 가지 정보로 구성됩니다.

- typ: 토큰의 타입을 지정합니다.
- alg: 알고리즘 방식을 지정합니다. (서명 및 토큰 검증에 사용)

```
{ "typ": "JWT", "alg": "HS256" }
```

Payload

토큰의 Payload에는 토큰에서 사용할 **정보의 조각들인 클레임(Claim)**이 담겨 있습니다.

클레임은 총 3가지로 나누어지며, JSON 형태로 다수의 정소를 넣을 수 있습니다.

Payload

등록된 클레임

토큰 정보를 표현하기 위해 **이미 정해진 종류의 데이터**입니다. 모두 선택적으로 작성이 가능하며 사용하는 것을 권장합니다.

Payload

공개 클레임

충돌이 방지된 이름을 가지고 있어야 합니다. 충돌을 방지하기 위해 클레임 이름을 URI 형식으로 짓습니다.

```
{ https://elice.io: true }
```

Payload

비공개 클레임

등록과 공개를 제외한 클레임이자 사용자 정의 클레임으로 서버와 클라이언트 협의로 사용되는 클레임입니다. **서버와 클라이언트 사이에 임의로 지정한 정보를 저장**합니다. 이 클레임은 공개 클레임과 달리 이름이 중복되어 충돌될 수 있으니 유의해야 합니다.

```
{ "student_name": "elice" }
```

Signature

JWT의 마지막 부분인 서명은 **토큰을 인코딩하거나 유효성 검증을 할 때 사용하는 고유한 암호화 코드**입니다.

서명은 헤더의 인코딩 값과 정보의 인코딩 값을 합친 후 주어진 비밀키로 해시를 하여 생성합니다.

크레딧

/* elice */

코스 매니저
이재성

콘텐츠 제작자
이바울, 이재성

강사
이바울

감수자
이바울

디자이너
강혜정

연락처

TEL
070-4633-2015

WEB
https://elice.io

E-MAIL
contact@elice.io